

AMS Emulation Technology

Results From DARPA POSH – Phase 1

Chris Tice, VP Verification Continuum Solutions and Manufacturing

Oleg Raikhman, Solutions Architect

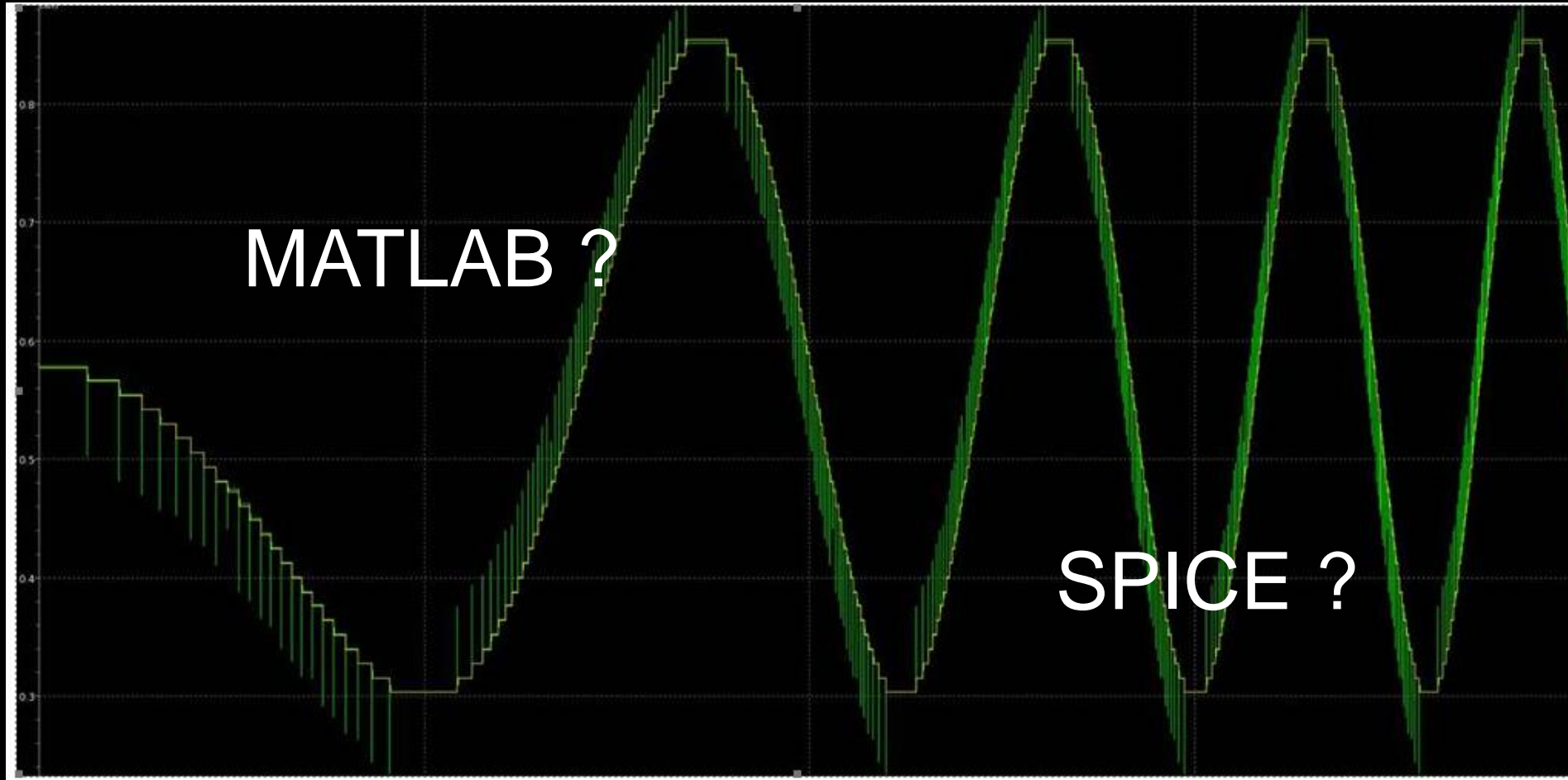
Yishai Statter, R&D Engineer

July 2019

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government."



What is this?

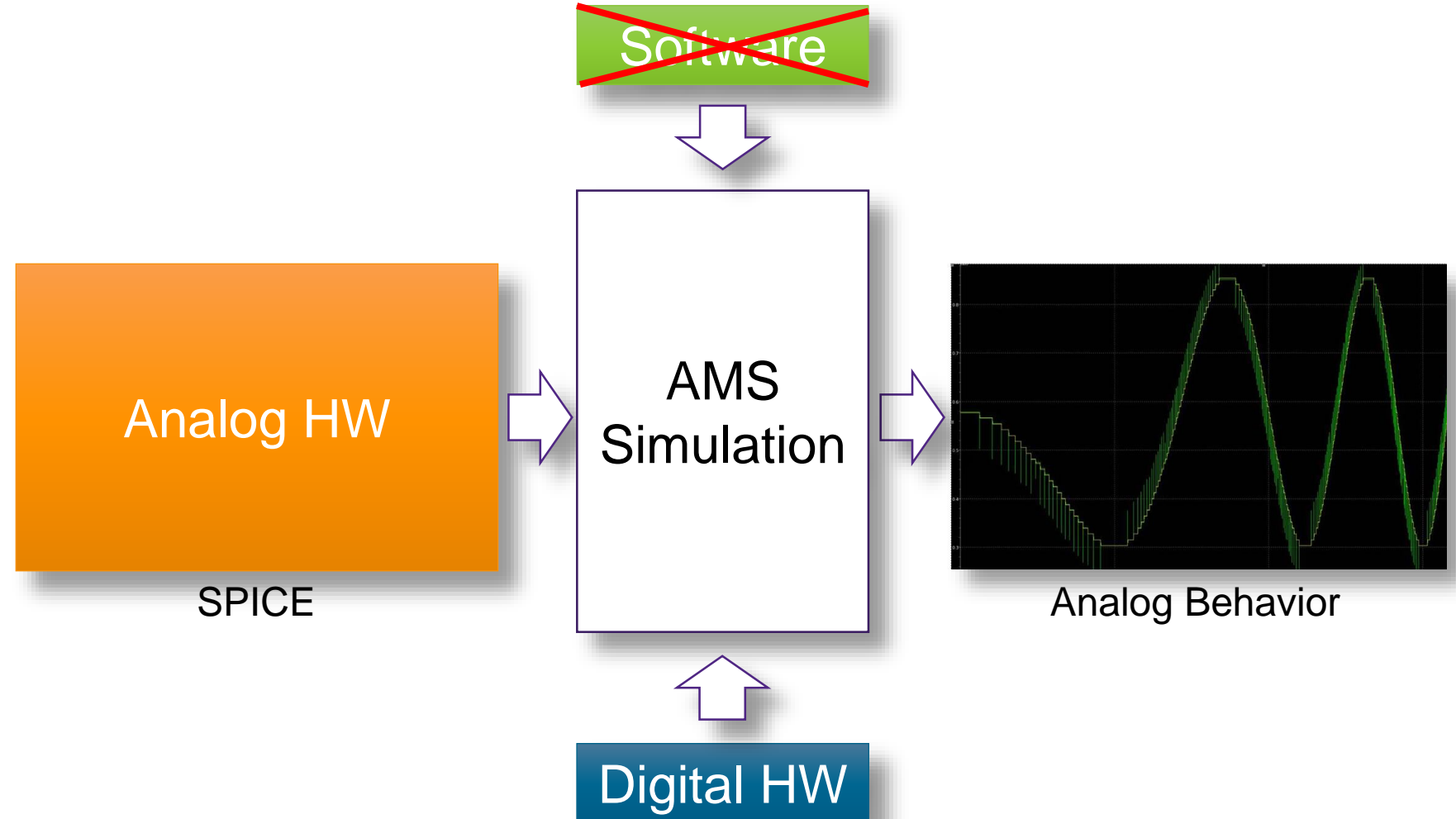


Or

AMS Verification Today

Too Slow for Software and System Validation

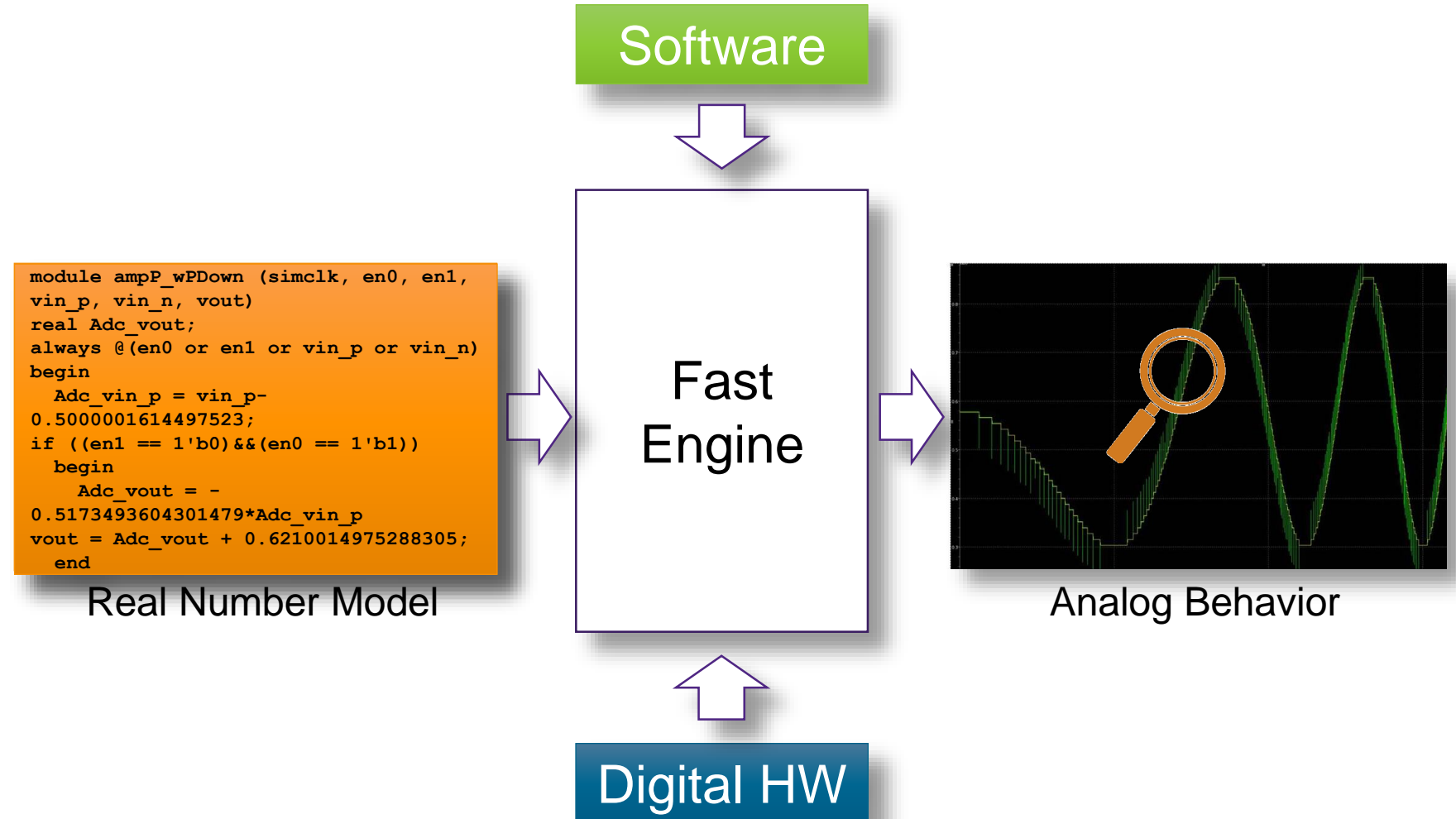
- Engine
 - Analog Mixed-Signal Simulation (SPICE + SV)
- Advantage
 - Accuracy
- Disadvantage
 - Slow



What's Needed

System and Software Validation for AMS Designs

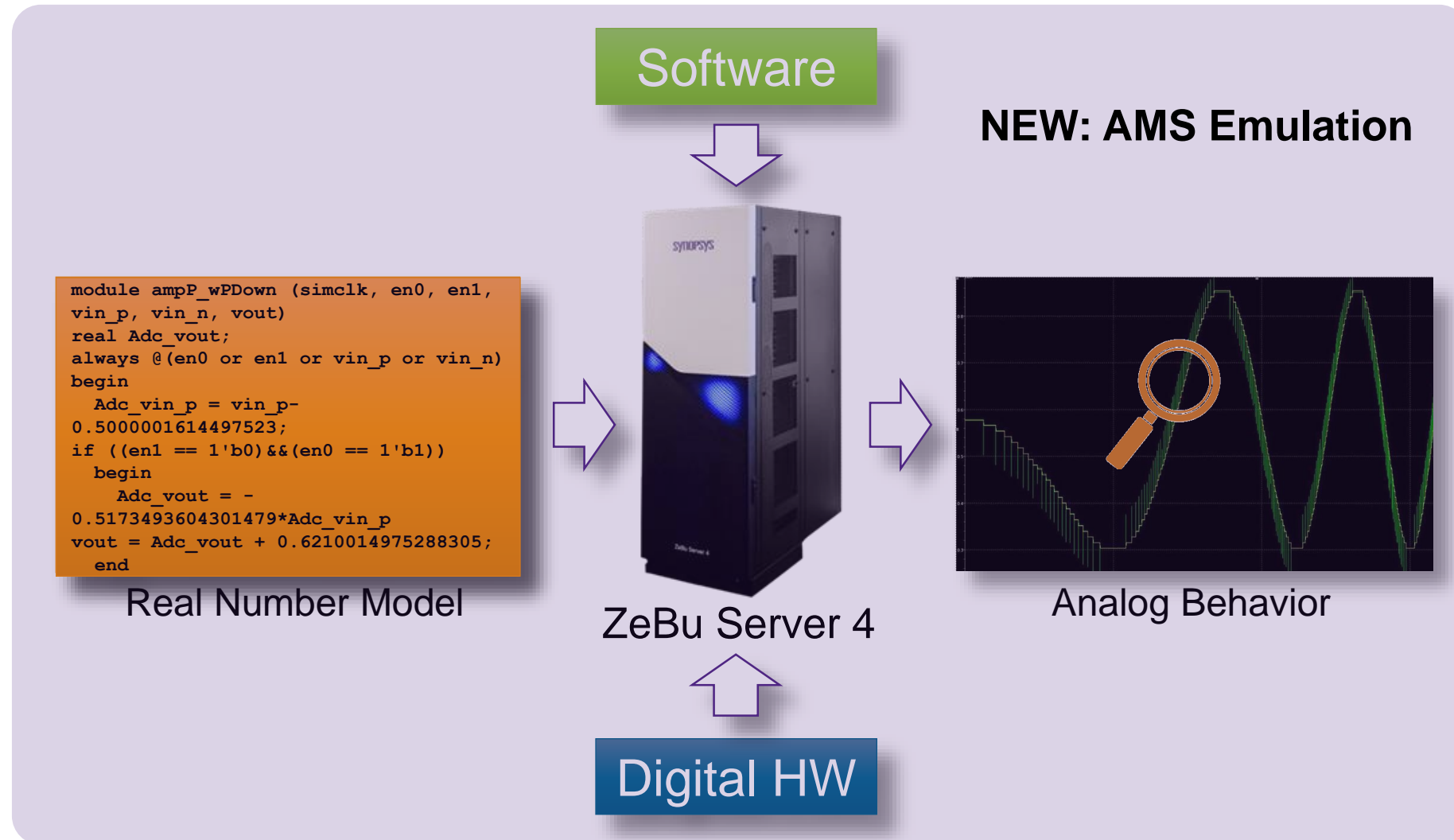
- Engine
 - Ability to execute software workloads
- Modeling
 - Using standard language
- Accuracy
 - Sufficient for system level integration analysis



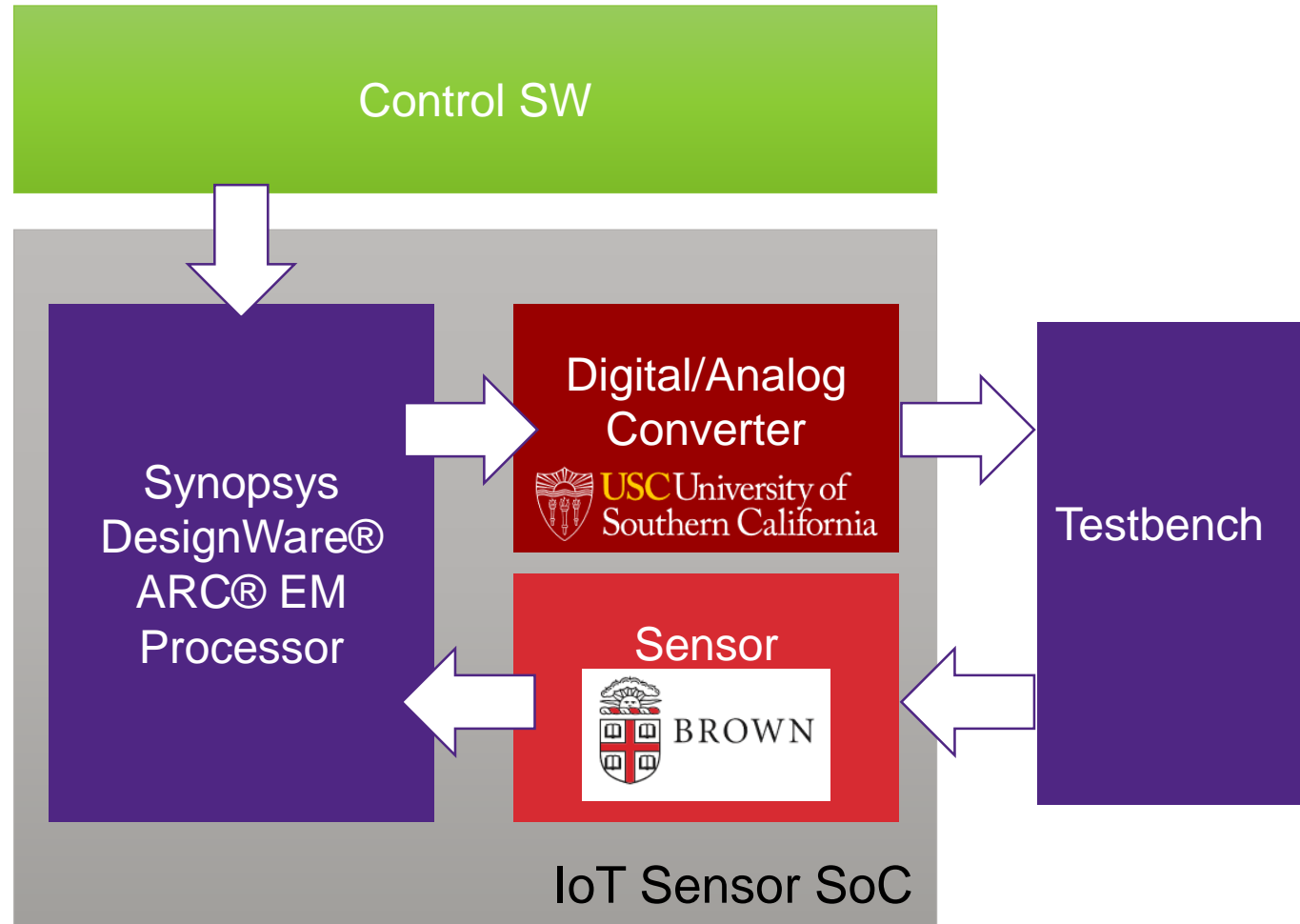
DARPA POSH – Phase 1 Result: AMS Emulation

System and Software Validation for AMS Designs

- Engine
 - Emulation
- Modeling
 - Real Number Model using SV data type
- Accuracy
 - Sufficient for system level integration analysis



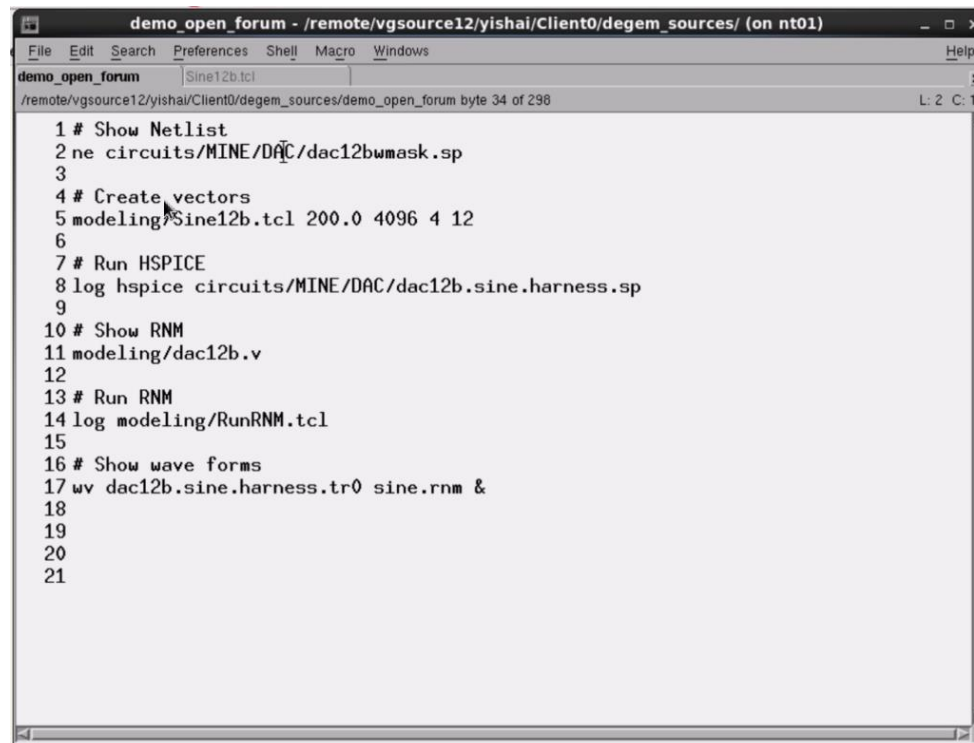
Demo Time: Let's Take a Quick Look



SPICE Conversion to RNM Demonstration (1)

FLOW

We will display the input netlist (a mix of digital and analog content), show how the tool recognizes and separates this content, run SPICE, convert the netlist to RNM and display it (takes only 1 second to simulate RNM vs. 70 for SPICE), and compare the simulation output of SPICE input against RNM output

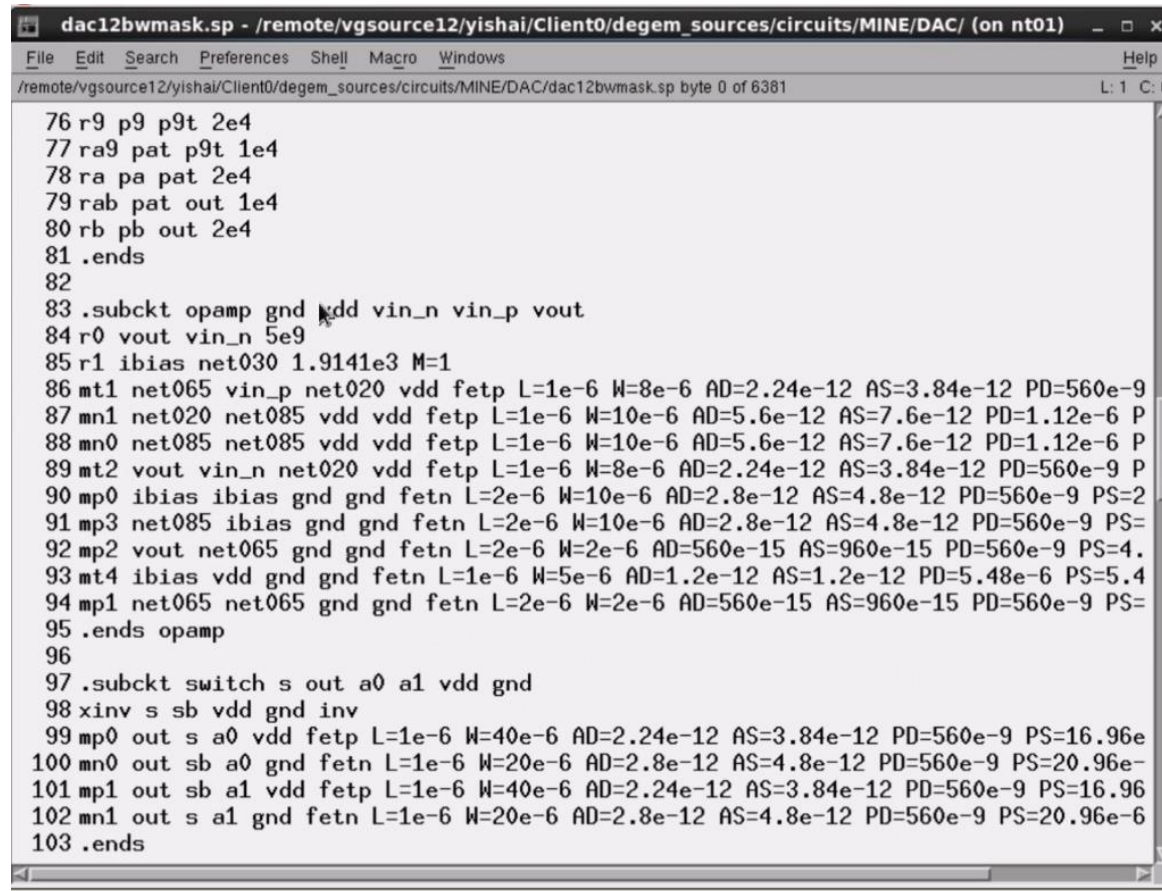


```
demo_open_forum - /remote/vgsource12/yishai/Client0/degem_sources/ (on nt01)
File Edit Search Preferences Shell Macro Windows Help
demo_open_forum Sine12b.tcl
/remote/vgsource12/yishai/Client0/degem_sources/demo_open_forum byte 34 of 298 L: 2 C: 19

1 # Show Netlist
2 ne circuits/MINE/DAC/dac12bmask.sp
3
4 # Create vectors
5 modeling/Sine12b.tcl 200.0 4096 4 12
6
7 # Run HSPICE
8 log hspice circuits/MINE/DAC/dac12b.sine.harness.sp
9
10 # Show RNM
11 modeling/dac12b.v
12
13 # Run RNM
14 log modeling/RunRNM.tcl
15
16 # Show wave forms
17 wv dac12b.sine.harness.tr0 sine.rnm &
18
19
20
21
```


SPICE Conversion to RNM Demonstration (2)

Snippet of DAC SPICE netlist

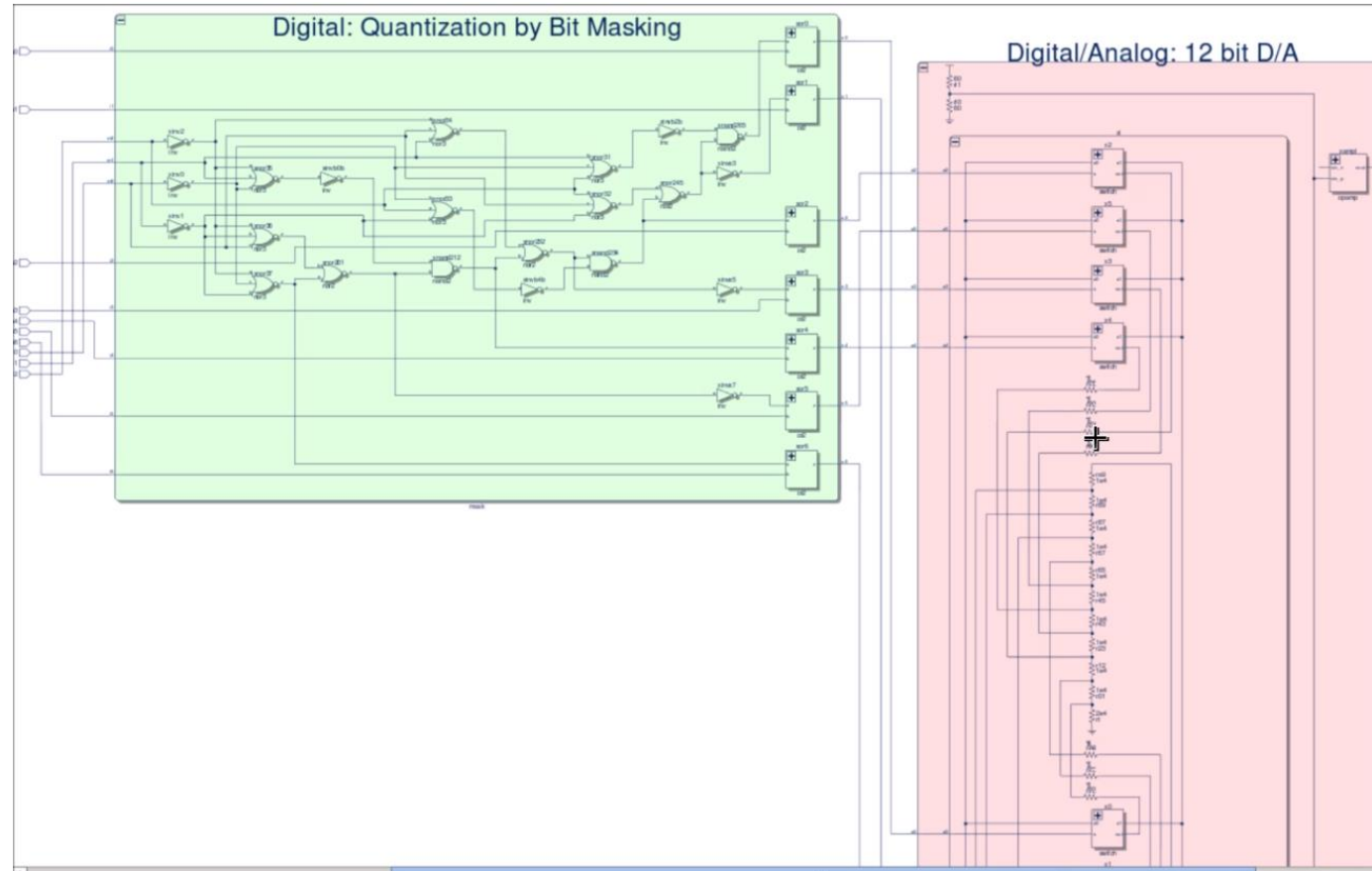


The screenshot shows a text editor window titled "dac12bwmask.sp - /remote/vgsource12/yishai/Client0/degem_sources/circuits/MINE/DAC/ (on nt01)". The window contains a SPICE netlist for a DAC circuit. The netlist includes component definitions, node connections, and model parameters for various transistors and opamps. The code is as follows:

```
76 r9 p9 p9t 2e4
77 ra9 pat p9t 1e4
78 ra pa pat 2e4
79 rab pat out 1e4
80 rb pb out 2e4
81 .ends
82
83 .subckt opamp gnd vdd vin_n vin_p vout
84 r0 vout vin_n 5e9
85 r1 ibias net030 1.9141e3 M=1
86 mt1 net065 vin_p net020 vdd fetp L=1e-6 W=8e-6 AD=2.24e-12 AS=3.84e-12 PD=560e-9
87 mn1 net020 net085 vdd vdd fetp L=1e-6 W=10e-6 AD=5.6e-12 AS=7.6e-12 PD=1.12e-6 P
88 mn0 net085 net085 vdd vdd fetp L=1e-6 W=10e-6 AD=5.6e-12 AS=7.6e-12 PD=1.12e-6 P
89 mt2 vout vin_n net020 vdd fetp L=1e-6 W=8e-6 AD=2.24e-12 AS=3.84e-12 PD=560e-9 P
90 mp0 ibias ibias gnd gnd fetn L=2e-6 W=10e-6 AD=2.8e-12 AS=4.8e-12 PD=560e-9 PS=2
91 mp3 net085 ibias gnd gnd fetn L=2e-6 W=10e-6 AD=2.8e-12 AS=4.8e-12 PD=560e-9 PS=
92 mp2 vout net065 gnd gnd fetn L=2e-6 W=2e-6 AD=560e-15 AS=960e-15 PD=560e-9 PS=4.
93 mt4 ibias vdd gnd gnd fetn L=1e-6 W=5e-6 AD=1.2e-12 AS=1.2e-12 PD=5.48e-6 PS=5.4
94 mp1 net065 net065 gnd gnd fetn L=2e-6 W=2e-6 AD=560e-15 AS=960e-15 PD=560e-9 PS=
95 .ends opamp
96
97 .subckt switch s out a0 a1 vdd gnd
98 xinv s sb vdd gnd inv
99 mp0 out s a0 vdd fetp L=1e-6 W=40e-6 AD=2.24e-12 AS=3.84e-12 PD=560e-9 PS=16.96e
100 mn0 out sb a0 gnd fetn L=1e-6 W=20e-6 AD=2.8e-12 AS=4.8e-12 PD=560e-9 PS=20.96e-
101 mp1 out sb a1 vdd fetp L=1e-6 W=40e-6 AD=2.24e-12 AS=3.84e-12 PD=560e-9 PS=16.96
102 mn1 out s a1 gnd fetn L=1e-6 W=20e-6 AD=2.8e-12 AS=4.8e-12 PD=560e-9 PS=20.96e-6
103 .ends
```


SPICE Conversion to RNM Demonstration (3)

SPICE conversion tool recognizes and separates digital and analog circuits



SPICE Conversion to RNM Demonstration (4)

SPICE simulation takes 70 seconds (RNM takes 1 second)

```
File Edit View Search Terminal Help
# nodes      = 384 # elements = 128
# resistors  = 31 # capacitors = 1 # inductors = 0
# mutual_inds = 0 # vccs = 0 # vcvs = 0
# cccs = 0 # ccvs = 0 # volt_srcs = 15
# curr_srcs = 0 # diodes = 0 # bjts = 0
# jfets = 0 # mosfets = 81 # U elements = 0
# T elements = 0 # W elements = 0 # B elements = 0
# S elements = 0 # P elements = 0 # va device = 0
# vector_srcs = 0 # N elements = 0

***** Runtime Statistics (seconds) *****

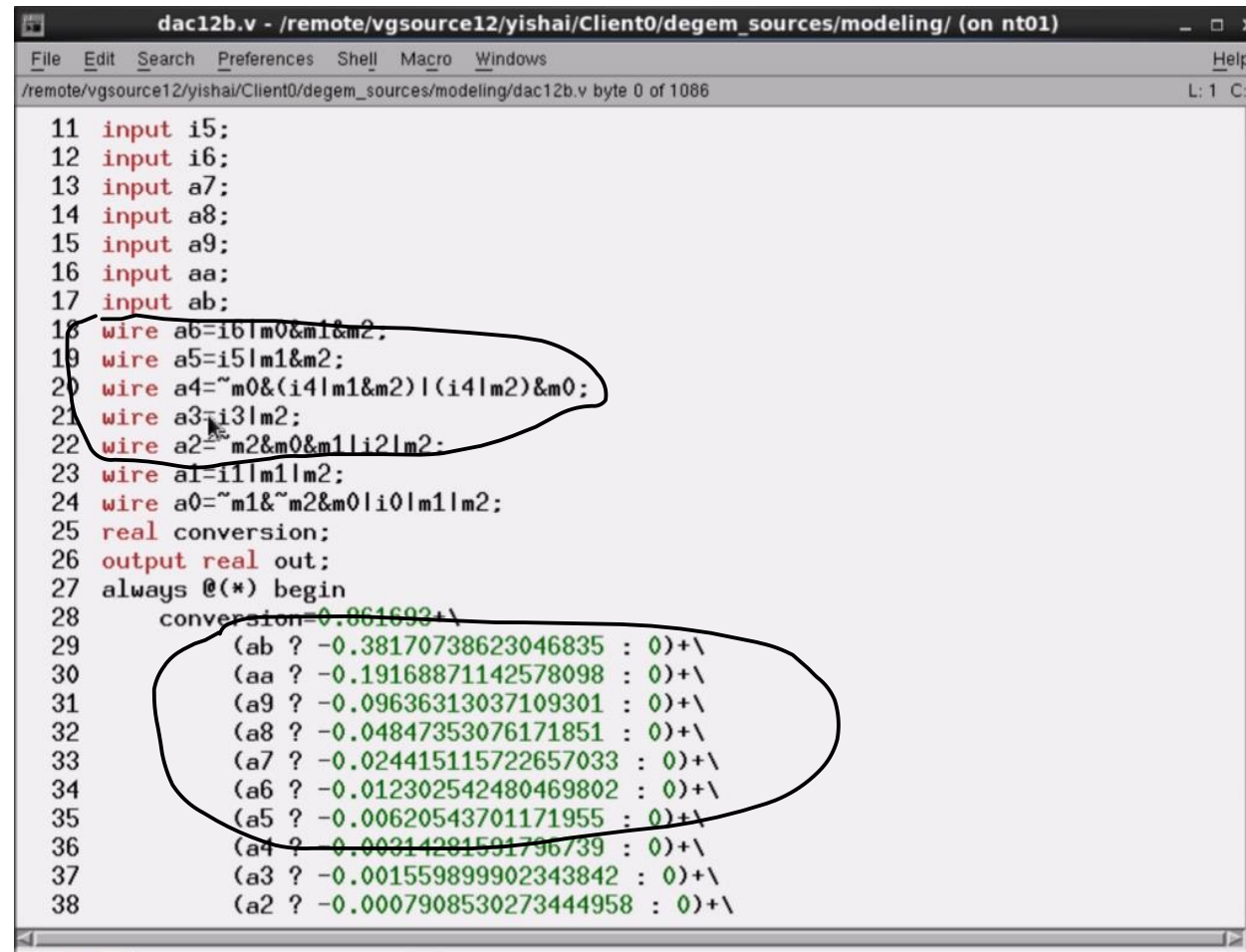
analysis      time  # points  tot. iter  conv.iter
op point      0.01      1        26
transient     68.52    20001    319429    220492 rev= 15005
readin        0.72
errchk        0.07
setup         0.01
output        0.00

peak memory used 556.14 megabytes
total cpu time 69.@DURATION seconds
total elapsed time 70.@DURATION seconds
job started at 14:28:00 06/27/2019
job ended at 14:29:10 06/27/2019

>info: ***** hspice job concluded
lic: Release hspice token(s)
lic: total license checkout elapse time: 0.74(s)
Thu Jun 27 14:29:10 EDT 2019
Total run time: 70 seconds
DECSM_ROOT>
```

SPICE Conversion to RNM Demonstration (5)

Snippet of RNM generated (note mix of digital Boolean and analog real numbers)

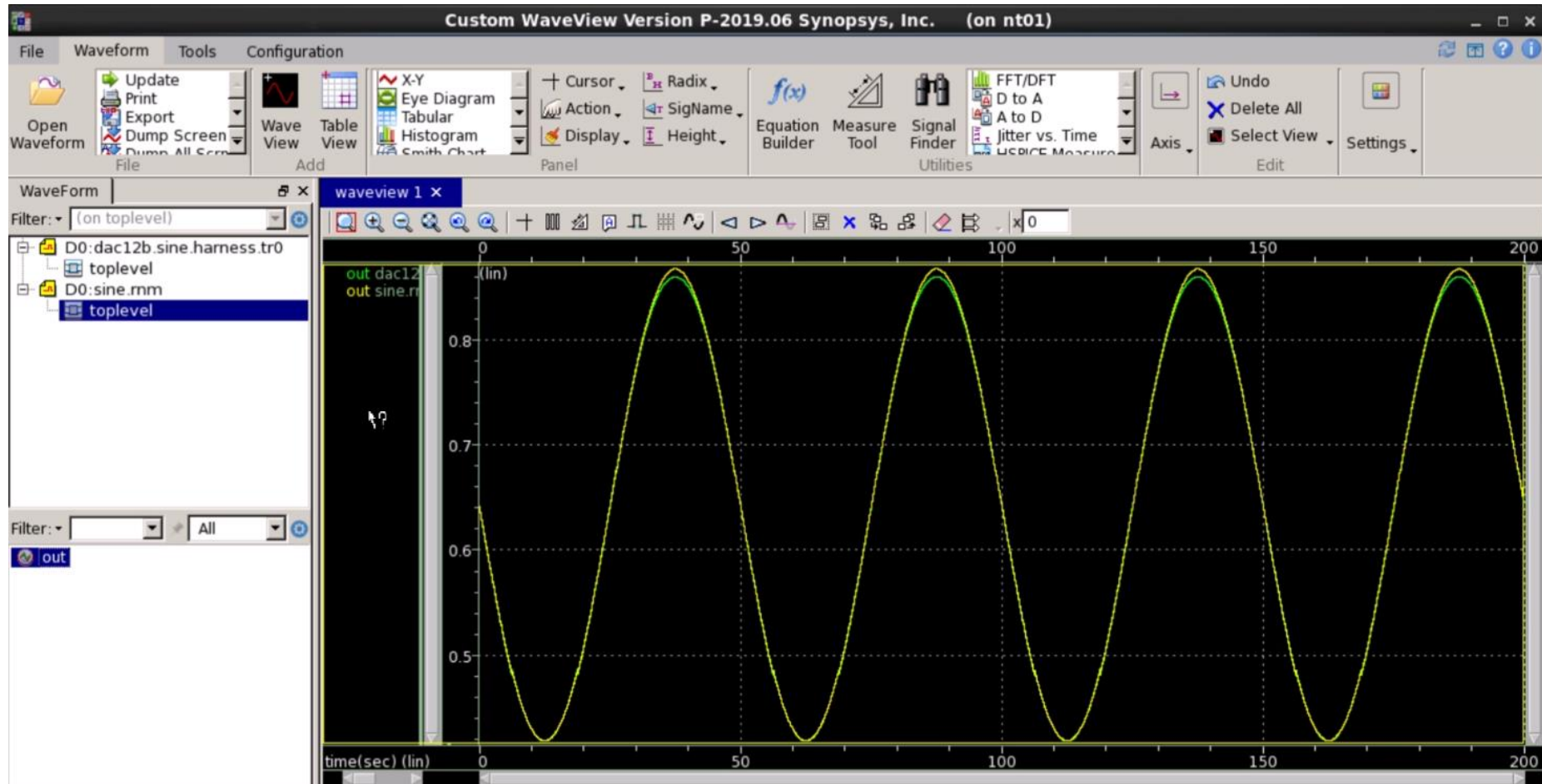


```
dac12b.v - /remote/vgsource12/yishai/Client0/degem_sources/modeling/ (on nt01)
File Edit Search Preferences Shell Macro Windows Help
/remote/vgsource12/yishai/Client0/degem_sources/modeling/dac12b.v byte 0 of 1086 L: 1 C: 0

11 input i5;
12 input i6;
13 input a7;
14 input a8;
15 input a9;
16 input aa;
17 input ab;
18 wire a6=i6|m0&m1&m2;
19 wire a5=i5|m1&m2;
20 wire a4=~m0&(i4|m1&m2)|(i4|m2)&m0;
21 wire a3=i3|m2;
22 wire a2=m2&m0&m1|i2|m2;
23 wire a1=i1|m1|m2;
24 wire a0=~m1&~m2&m0|i0|m1|m2;
25 real conversion;
26 output real out;
27 always @(*) begin
28     conversion=0.061693+
29     (ab ? -0.38170738623046835 : 0)+\
30     (aa ? -0.19168871142578098 : 0)+\
31     (a9 ? -0.09636313037109301 : 0)+\
32     (a8 ? -0.04847353076171851 : 0)+\
33     (a7 ? -0.024415115722657033 : 0)+\
34     (a6 ? -0.012302542480469802 : 0)+\
35     (a5 ? -0.00620543701171955 : 0)+\
36     (a4 ? -0.00314281591796739 : 0)+\
37     (a3 ? -0.001559899902343842 : 0)+\
38     (a2 ? -0.0007908530273444958 : 0)+\
```

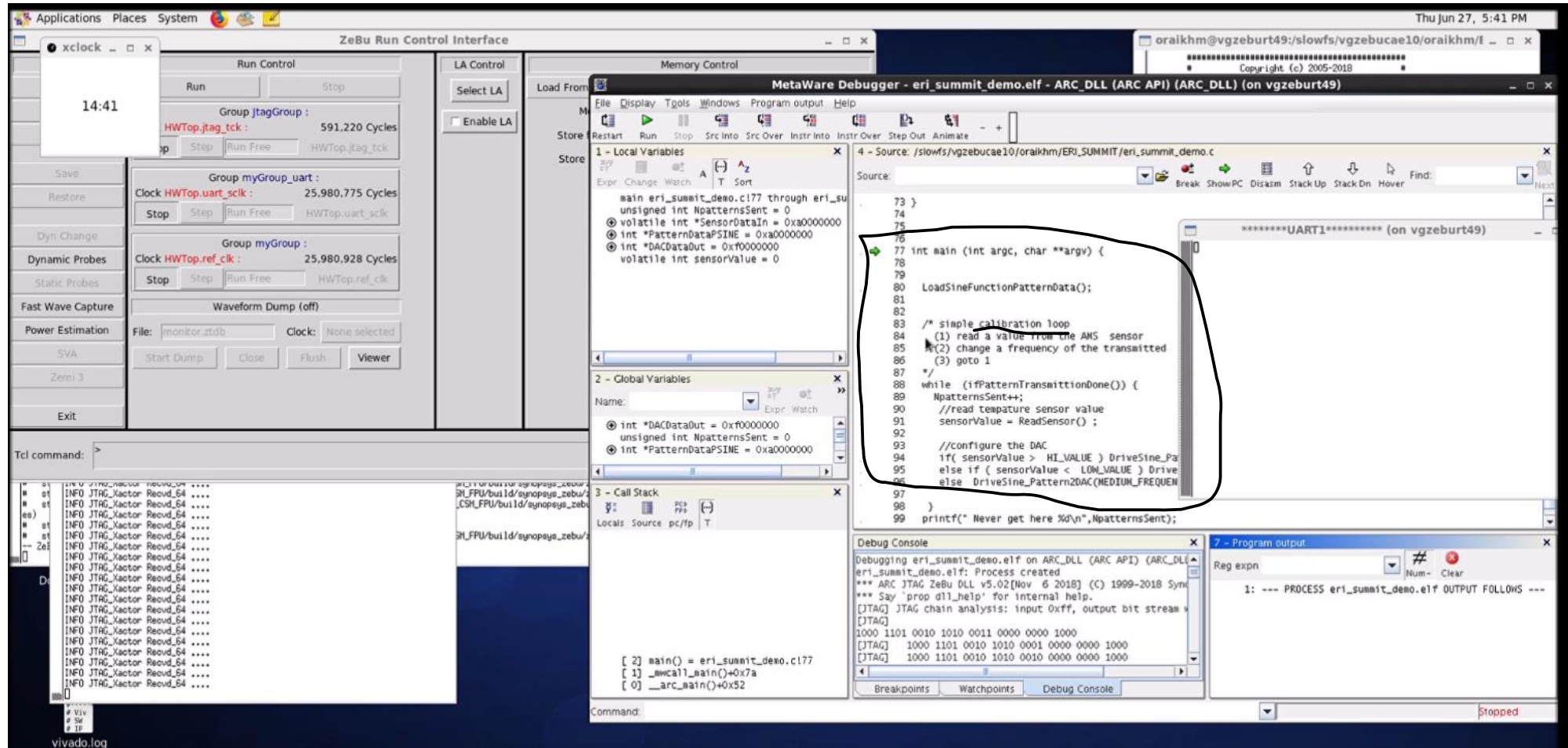
SPICE Conversion to RNM Demonstration (6)

Comparison of SPICE vs. RNM simulation output waveforms



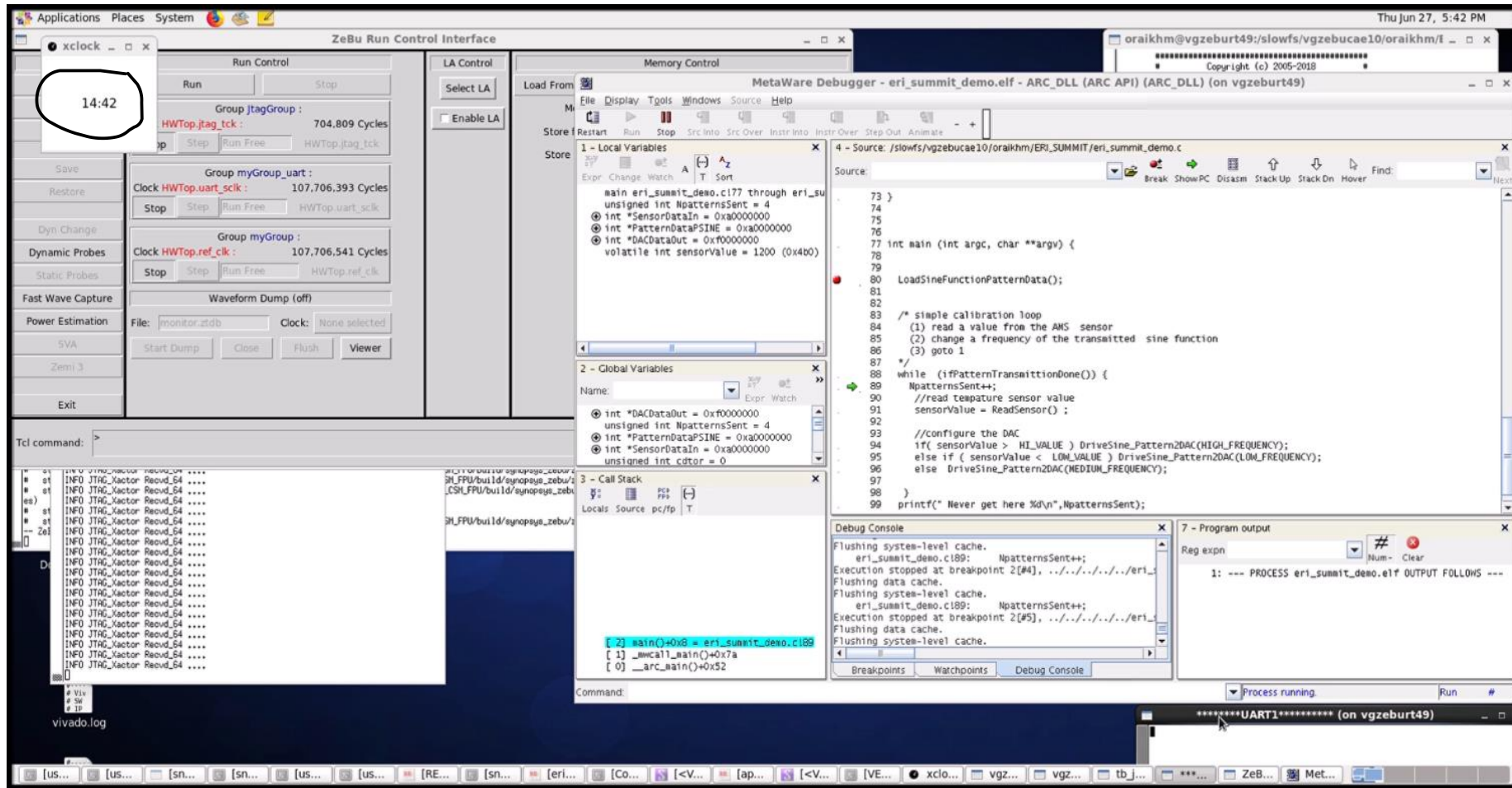
AMS Emulation (1)

Debugger at breakpoint showing calibration software running on ZeBu emulator



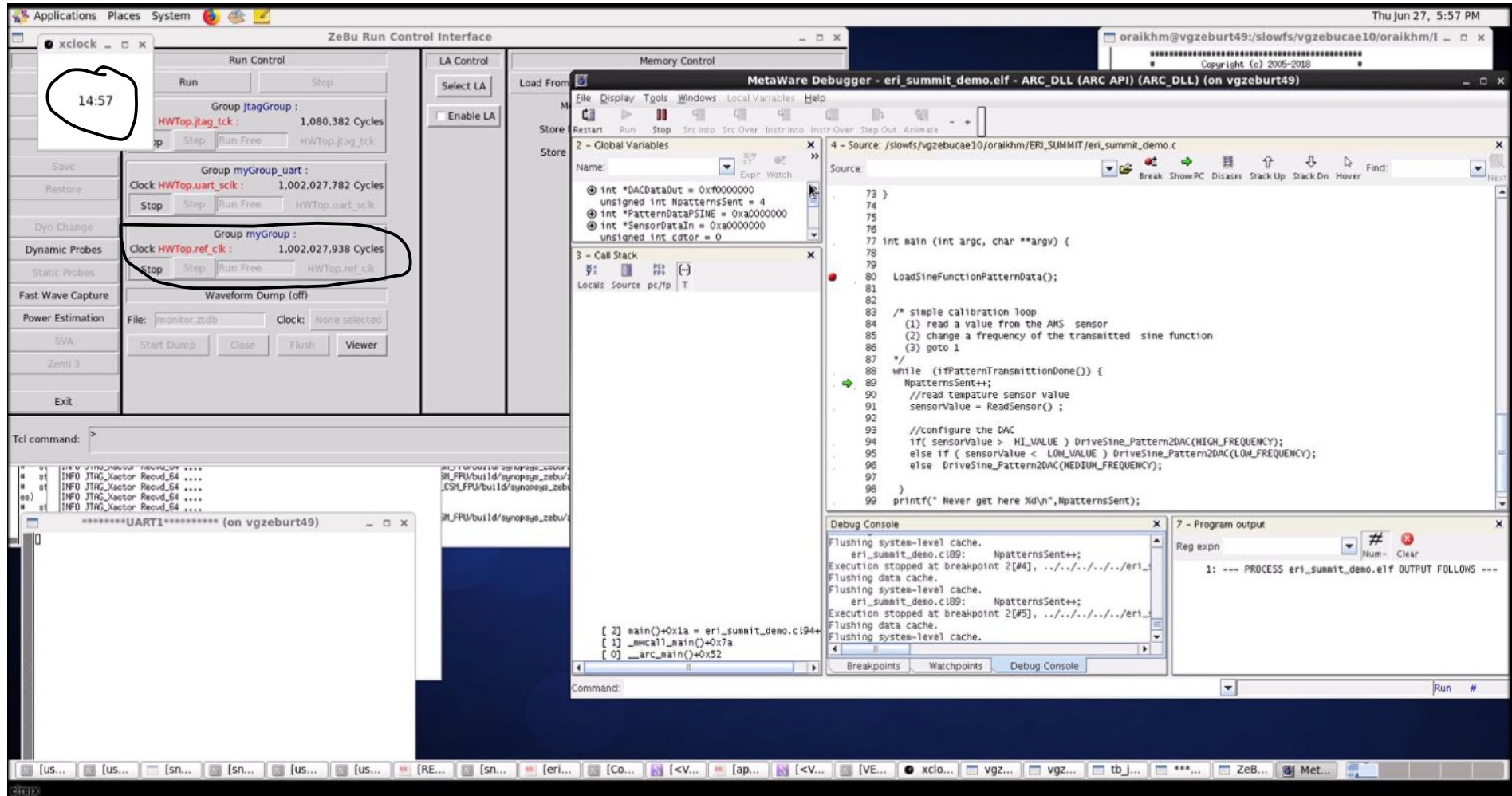
AMS Emulation (2)

At 2:42 PM, remove breakpoint and let emulator run (@ 2 MHz)



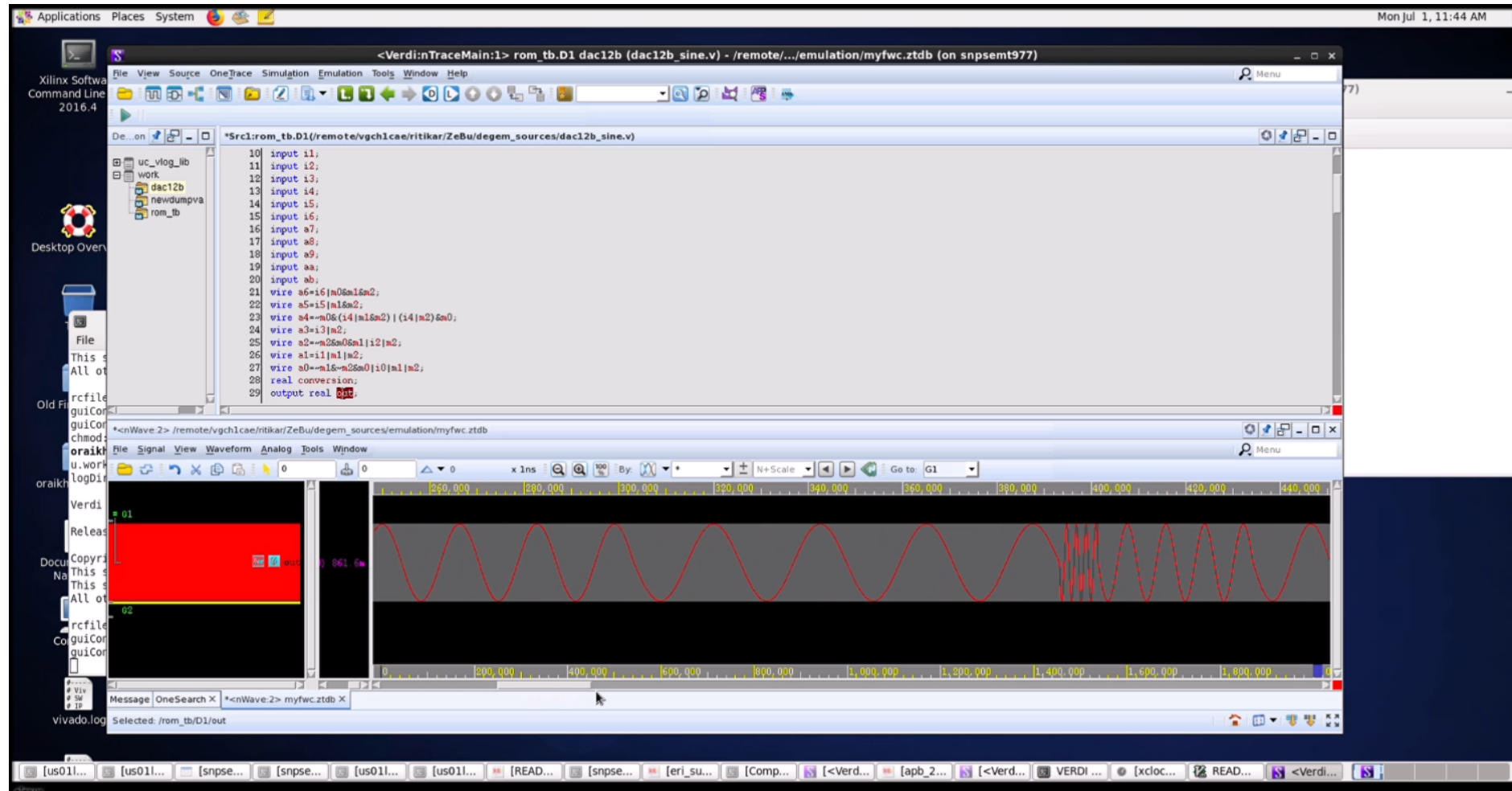
AMS Emulation (3)

At 2:57 PM, 15 seconds later, have already run over one billion cycles



AMS Emulation (4)

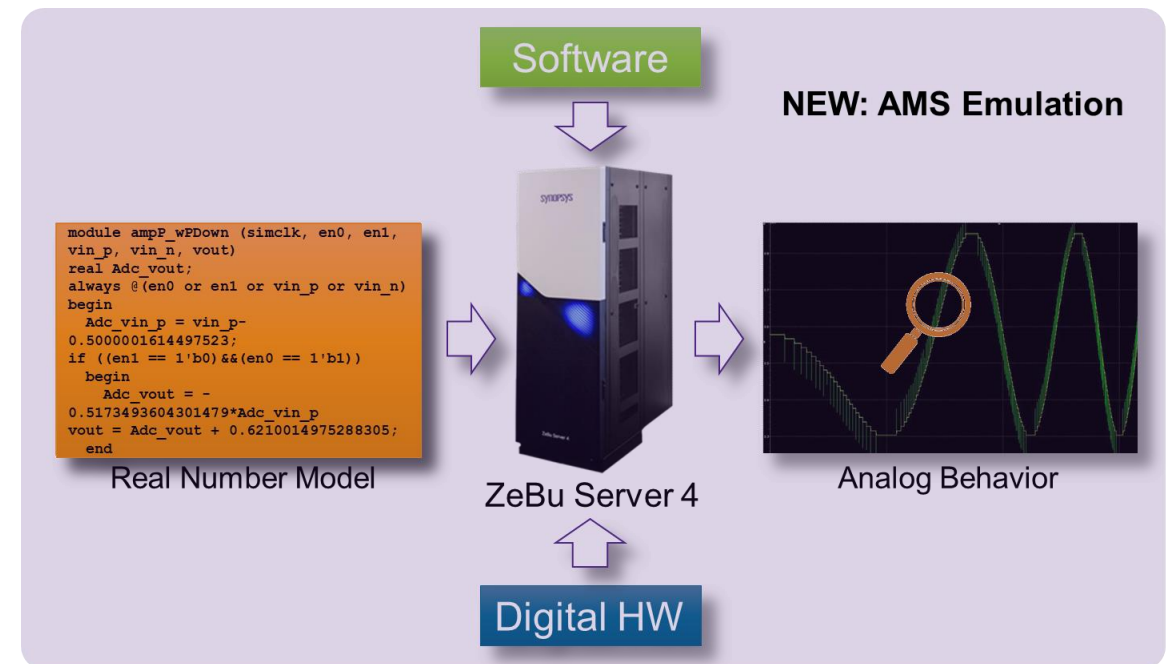
Analog waveform output from simulator shows circuit working



AMS Emulation: Summary and Next Steps

Results From DARPA POSH – Phase 1

- What you saw today
 - AMS emulation is working
 - AMS emulation delivers expected performance
 - Validation of AMS SoC including SW
- Next Steps
 - Continue to optimize AMS emulation technology
 - Enlarge the AMS emulation partner community
 - Develop complementary AMS assertions



Thank You

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

